PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 13/00

A1

(11) International Publication Number: WO 99/01819

(43) International Publication Date: 14 January 1999 (14.01.99)

(21) International Application Number: PCT/US98/13761

(22) International Filing Date: 30 June 1998 (30.06.98)

(30) Priority Data:
60/051,501 1 July 1997 (01.07.97) US

(71) Applicant: PROGRESS SOFTWARE CORPORATION [US/US]; 14 Oak Park, Bedford, MA 01730 (US).

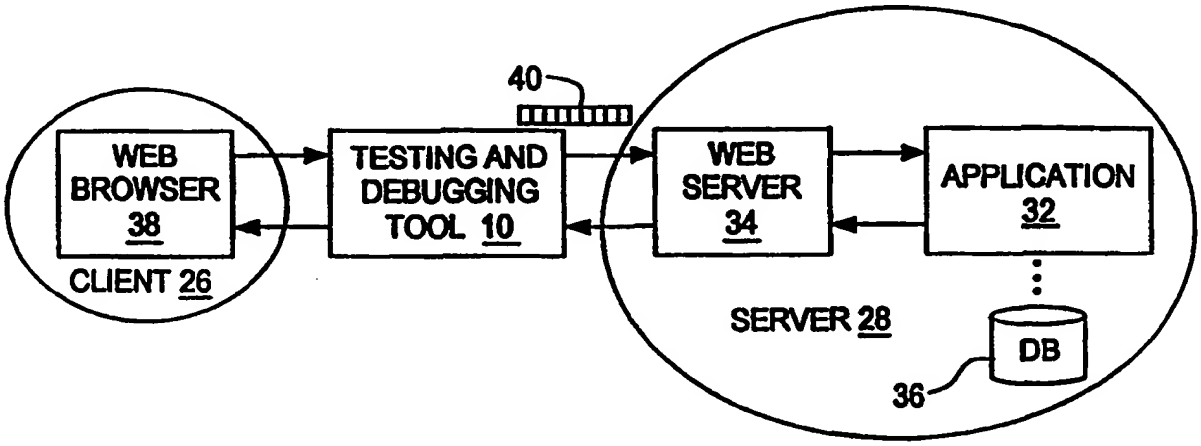
(72) Inventors: CASELLA, Stephen, R.; 8 Taconic Drive, Amherst, NH 03031 (US). BARDANI, Robert, L., Jr.; 2 Royal Crest Drive #12, Nashua, NH 03060 (US). SWAN, David, M.; Apartment 315, 1 Clocktower Place, Nashua, NH 03060 (US).

(74) Agents: SCHURGIN, Stanley, M. et al.; Weingarten, Schurgin, Gagnebin & Hayes LLP, Ten Post Office Square, Boston, MA 02109 (US).

(81) Designated States: CA, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: TESTING AND DEBUGGING TOOL FOR NETWORK APPLICATIONS



(57) Abstract

A software development tool (10) permits capture, modification and recording of transactional messages that are transmitted between a client (26) and a server (28) in a computer network. A proxy is employed to capture messages such as requests and responses that are in transit between the client (26) and the server (28). The captured requests and responses can be displayed and modified before being retransmitted via the proxy. Further, transaction records can be selectively provided to at least one software application (46) for analysis.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

TITLE OF THE INVENTION

TESTING AND DEBUGGING TOOL FOR NETWORK APPLICATIONS

CROSS REFERENCE TO RELATED APPLICATIONS

Priority is claimed to U.S. Provisional Patent Application Serial No. 60/051,501 entitled TESTING AND DEBUGGING TOOL, filed July 1, 1997.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

Not Applicable.

BACKGROUND OF THE INVENTION

The present invention is related to software development tools, and more particularly to testing and debugging tools for network applications.

Testing and debugging tools that facilitate software development are known. However, known testing and debugging tools are generally not well suited for use with network applications such as internet web sites. Testing and debugging tools are typically operative with only one programming language. However, the software employed by an internet web site and browser may comprise a plurality of programming languages. Testing and debugging tools are designed to be employed prior to deployment of the application under development. However, the behavior of a network application following deployment in a "real" environment is often different than the behavior of the application in the development environment.

BRIEF SUMMARY OF THE INVENTION

In accordance with the present invention, a software development tool permits capture, modification and recording of transactions between a client and a server in a computer network. The tool is situated in a communication path between the client and the server. A protocol-specific proxy is employed to capture data units that are associated with the transaction when the data units are transmitted between the client and the server. The transaction is displayed and optionally modified en route between the client and the server. Further, transaction records are selectively provided to at least one software application for analysis. The supported protocols may include TCP/IP protocols such as HTTP, FTP, SMTP, POP3 and IMAP4.

Selectively capturing and modifying data units between the client and the server enables useful features such as tracing and isolating transactions between the client and server. It is also possible to debug a deployed application, debug the internal interactions of a browser application by employing inbound and outbound streams, preview data units that the client device will receive from the server device, and set breakpoints and watch variables to selectively interrupt transactions. Advantages related to security and performance concerns about applets and components that instantiate in a browser application include enumeration of methods, fields and interfaces in a class prior to activation in the browser, enumeration of methods, properties and events in an ActiveX type library prior to activation in the browser, opening a CAB file prior to activation in the browser, and logging an "on-the-wire" transaction.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING

The foregoing features of this invention, as well as the invention itself, may be more fully understood from the

following Detailed Description of the Invention, and Drawing, of which:

Fig. 1 is a block diagram of the testing and debugging tool;

Fig. 2 is a block diagram that illustrates capture and modification of data units;

Fig. 3 is a block diagram that illustrates providing transaction records to selected applications;

Fig. 4 is a block diagram that illustrates use of the software development tool with an online shopping cart application;

Fig. 5 is a block diagram that illustrates use of the software development tool with a load testing application; and

Fig. 6 is a block diagram that illustrates use of the software development tool with an error testing application.

DETAILED DESCRIPTION OF THE INVENTION

U.S. Provisional Patent Application Serial No. 60/051,501 entitled TESTING AND DEBUGGING TOOL, filed July 1, 1997 is incorporated herein by reference.

Referring to Fig. 1, a testing and debugging tool 10 for network applications includes a main application 12 and a transaction logging application 14. The main application contains an editor 15 and a plurality of TCP/IP protocol proxies including an HTTP proxy 16, an FTP proxy 18, an SMTP proxy 20, a POP3 proxy 22 and an IMAP4 proxy 24 in the illustrated embodiment.

Referring to Fig. 2, the testing and debugging tool 10 is situated in a communication path between a selected client device 26 and a selected server device 28. The entire communication path, including the client 26, server 28 and testing and debugging tool 10, may exist on a single computer device or, as illustrated, on multiple computer devices and interconnecting media. The server device 28 is selected by specifying a DNS name or IP address that is associated with

the server device. The client device 26 is selected by specifying a DNS name or IP address that is associated with the client device. Alternatively, the source address of the client can be snooped from a request 29 that is transmitted from the client 26 to the server 28 via the testing and debugging tool 10. In a connection between the client 26 and the server 28, the IP address of the testing and debugging tool 10 is employed for communications from the client 16 to the testing and debugging tool 10, and the IP address of the server 28 is employed for communication from the testing and debugging tool 10 to the server 28. Similarly, the IP address of the testing and debugging tool 10 is employed for communications from the server 28 to the testing and debugging tool 10, and the IP address of the client 26 is employed for communications from the testing and debugging tool 10 to the client 26. The DNS name or IP address of the server 28 is not required for the HTTP protocol. A destination identifier is extracted from the requested URL when HTTP is employed. A specified 32-bit signed numeric value indicates the port which the testing and debugging tool monitors for a connection from the client 26.

In a passive mode, transactions are monitored by the testing and debugging tool 10 without interrupting transmission between the client 26 and the server 28. The monitoring function may include display of transaction records. In the illustrated embodiment, a representation of requests 29, responses 31, and both the IP address and DNS name that are being employed by the active proxy are displayed.

Transaction requests 29 and responses 31 can also be captured and modified in the passive mode. In particular, the captured data units may be modified and then transmitted to the original destination. In the illustrated embodiment, request 29 would be modified to provide request 33, which is transmitted to server 28. Similarly, response 31 would be modified to provide response 35, which is transmitted to client 26. The testing and debugging tool may be equipped

with an editor to facilitate modification of transactional data. Further, modifications can be automated for operation upon multiple transactions.

A breakpoint condition can be specified under which the tool will interrupt transactions. In the illustrated example, transactions such as requests and responses are monitored and logged in the passive mode until a breakpoint condition is satisfied. The tool then enters a break mode in which the triggering transaction is interrupted. The request or response that comprises the triggering transaction is then presented to the user for viewing and editing. The user can choose to remain in the break mode and intercept subsequent requests and responses, or exit the break mode and monitor transactions until another breakpoint condition occurs. If the breakpoint occurs during a request, a response can be composed with the tool and sent to the client, thereby circumventing the server. Breakpoints can be set to trigger upon receipt of a request, a response, a specified request method, a response to a specified request method, a request made to a specified host (a specified request or every request), a request made for a specified URL (a specified request or every request), a specified status code (in response to any method or to a specified method), a response containing a status code within a specified category (in response to any method or to a specified method), a specified message header present in a request or a response, a message header with a specified value present in a request or a response, a specified HTTP version in a request or a response, a malformed HTTP request, and a malformed HTTP response.

Stream filters 37 can be employed to restrict the flow of information between the client 26 and the server 28. When a filter is set, only transactions that satisfy the filter criteria are forwarded from the proxy to the specified destination. Transactions that do not satisfy the filter criteria are filtered out. For example, an HTTP filter such as "Transaction Type = GET AND File Type = IMAGE AND File

Size > 20000" filters out GET requests for any image file that is larger than 20,000 bytes. A message is displayed to indicate that the GET request had been made, but the request is not forwarded to the server 28. A POP3 filter such as "If Transaction Type = RETR Then Discard Attachments" strips any MIME or UU encoded attachments from incoming mail messages, and sends the text portion of the message to the mail application associated with the client. Stream filters 37 can be set based on any combination of file type, file size, file date/time, breakpoint criteria, and View Filters.

View filters 39 can be employed to control the amount of information that is displayed during monitoring. View filters 39 only limit what is displayed, and do not affect the flow of data between the client 26 and the server 28. For example, an HTTP view filter such as "Transaction Type = POST" will filter everything except POST requests from being displayed. View filters 39 are defined with the same command syntax that is used to create stream filters.

Referring to Fig. 3, copies of transactions are selectively provided to at least one software application 30 in an active mode. In particular, the transaction logging application 14 (Fig. 1) maintains a record of each transaction that is captured by the active proxy. The transaction record is maintained at least until a copy of the transaction record is transmitted to a predetermined application 30. Transaction records can be sent to multiple applications if desired. The applications perform functions such as analysis based on the transaction records.

Fig. 4 illustrates an implementation of the testing and debugging tool 10 for analysis of an online store application 32 that employs a "shopping cart." The online store application 32 is associated with a web server application 34 and a database 36, both of which are associated with server device 28. A web browser 38 operating on client device 26 is employed to access the online store. The testing and debugging tool 10 is situated in the communication path between the web browser 38 and the web

server 34. The shopping cart includes a record of items that have been selected for purchase via the browser 38. It is desirable to maintain a record of information associated with the transactions between the browser 38 and the web server 34, i.e., "state awareness," so that the status of the connection can be restored in the event that the connection between the web browser and the web server fails. For example, it is desirable to have a record of which items were selected for purchase before the connection to the online store fails. It is known to employ a "cookie" 40 to maintain state awareness. The cookie 40 is an encoded string that is generated by the online store application 32 and transmitted to the web browser 38. The cookie indicates state information. The web browser 38 stores the cookie 40. If the connection fails, the web browser contains the cookie 40 received from the online store application 32 when the connection is restored. The cookie is employed by the application 32 to restore the pre-failure state of the client/server session.

In the illustrated example, the testing and debugging tool 10 is employed to test the cookie 40 during development of the online store application 32. In particular, the cookie 40 is viewed en route from the web server 34 to the web browser 38, and also en route from the web browser to the web server. Further, the cookie can be modified en route without modifying the source code of the online store application 32.

Fig. 5 illustrates an implementation of the testing and debugging tool 10 for facilitating analysis of the performance of a web server 42. The testing and debugging tool 10 is situated in a communication path between a web browser 44 and the web server 42. A load testing application 46 is coupled to the testing and debugging tool 10. A request 48 is initiated at the web browser 44 and transmitted to the web server 42 via the testing and debugging tool 10. A response 50 to the request 48 is generated at the web server 42 and transmitted to the web browser 44 via the

testing and debugging tool 10. A record of the request 48 and the corresponding response 50, together the "transaction," is transmitted to the load testing application 46 from the testing and debugging tool 10. The load testing application 46 analyzes the amount of time required to provide the response 50. Further, the load testing application 46 can generate multiple requests, based on the initial request 48, to analyze the response generating performance of the web server 42 as the number of requests received by the web server is increased.

Fig. 6 illustrates use of the testing and debugging tool 10 with an error testing application 52. The testing and debugging tool is situated in a communication path between an FTP client 54 and an FTP server 56. The error detecting application 52 is coupled to the testing and debugging tool 10. Data is periodically transmitted from the FTP client 54 to the FTP server 56 for analysis and storage. A record of each transaction between the FTP client and the FTP server is provided to the error detecting application 52 by the testing and debugging tool 10. In the event that an error is detected by application 52, such as a failure in the connection between the FTP client and the FTP server, action is prompted, such as activating a pager device.

Having described the embodiments consistent with the present invention, other embodiments and variations consistent with the present invention will be apparent to those skilled in the art. Therefore, the invention should not be viewed as limited to the disclosed embodiments but rather should be viewed as limited only by the spirit and scope of the appended claims.

CLAIMS

What is claimed is:

1. A software tool that facilitates development of an application that operates in a computer network having a first device that transmits a data unit in a predetermined communication protocol toward a second device, comprising:

at least one proxy routine that is configured for operation with the predetermined communication protocol, and which is operative to capture the data unit that is formatted in said predetermined protocol and that is transmitted from said first device toward said second device; and

an editor that modifies said data unit after said data unit is captured, transmission of said data unit then being resumed toward the second device via said at least one proxy routine.

2. The software tool of claim 1 further including a routine that displays information including a representation of the data unit that is captured by the proxy routine.

3. The software tool of claim 2 further including a view filter that prevents specified information from being displayed.

4. The software tool of claim 1 further including a stream filter that modifies said data unit based upon predetermined criteria.

5. The software tool of claim 1 further including a trigger function that interrupts transmission between the first device and the second device upon detection of a predetermined operational condition.

6. The software tool of claim 1 further including a routine that prompts transmission of a representation of the data unit to at least one selected application that is coupled with the software tool.

7. A method for testing an application that operates on computer network having a first device and a second device, comprising the steps of:

transmitting a data unit from the first device toward the second device;

capturing the data unit with a proxy;

modifying the data unit; and

transmitting the modified data unit from the proxy toward the second device.

8. The method of claim 7 further including the step of displaying information that includes a representation of the data unit that is captured by the proxy.

9. The method of claim 8 further including the step of filtering the information prior to display.

10. The method of claim 7 further including the step of applying a filter to the data unit.

11. The method of claim 7 further including the step of interrupting transmission of the data unit between the first device and the second device upon the detection of a predetermined condition.

12. The method of claim 7 further including the step of providing a representation of the data unit that is captured by the proxy to at least one selected application.

13. A testing and debugging tool that facilitates development of a first application that operates on a first device in a computer network and is responsive to a second application that operates on a second device in the computer network, comprising:

at least one proxy that is associated with a predetermined network communication protocol and that is operative to receive a transaction request that are formatted

in said predetermined protocol and transmitted from the second application toward the first application; and

an editor that modifies said transaction request if desired and then prompts transmission of said transaction request toward said first application;

said at least one proxy being further operative to receive a transaction response that is formatted in said predetermined protocol and transmitted from the first application toward said second application in response to said transaction request,

said editor being operative to modify said transaction response if desired and prompt transmission of said transaction response toward said second application,

whereby the transaction comprising said request and said response is modifiable.

14. The software tool of claim 13 further including a routine that displays a representation of at least one of the transaction request and the transaction response.

15. The software tool of claim 14 further including a view filter that selectively filters information that is displayed.

16. The software tool of claim 13 further including a stream filter that selectively filters the transaction requests and responses.

17. The software tool of claim 13 further including a trigger that interrupts transmission between the first device and the second device upon the detection of a predetermined condition.

18. The software tool of claim 13 further including a routine that provides a representation of the transaction to a third application.

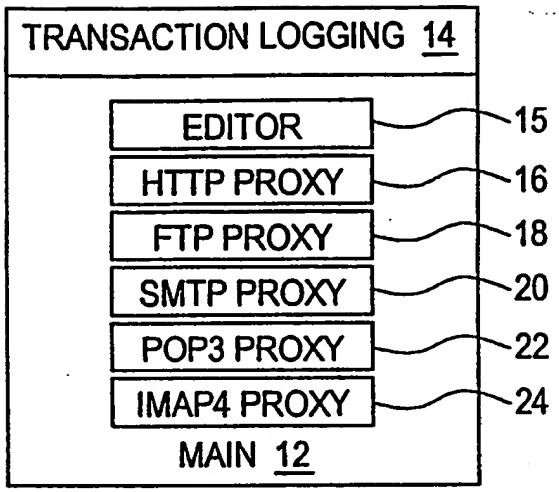


FIG. 1

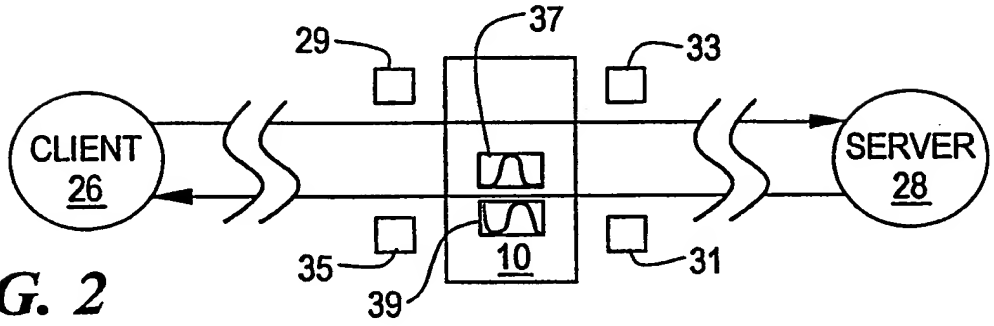


FIG. 2

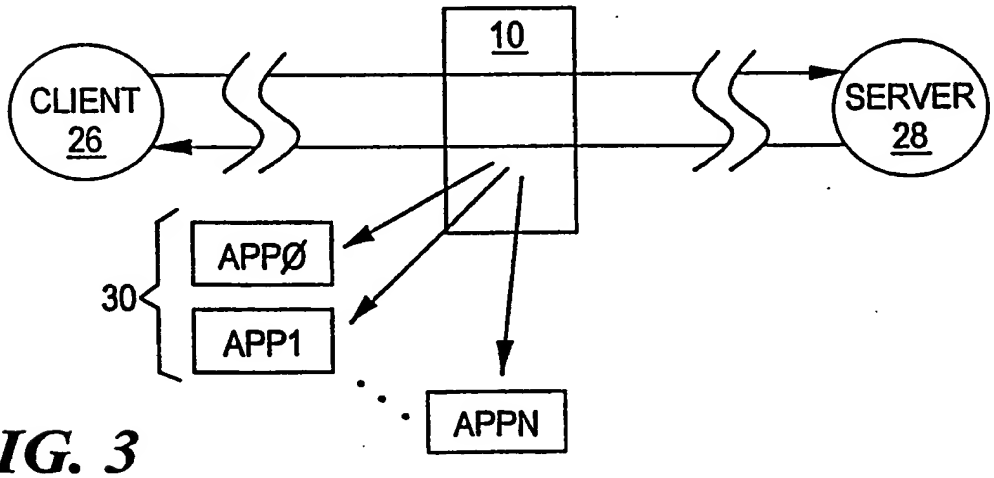


FIG. 3

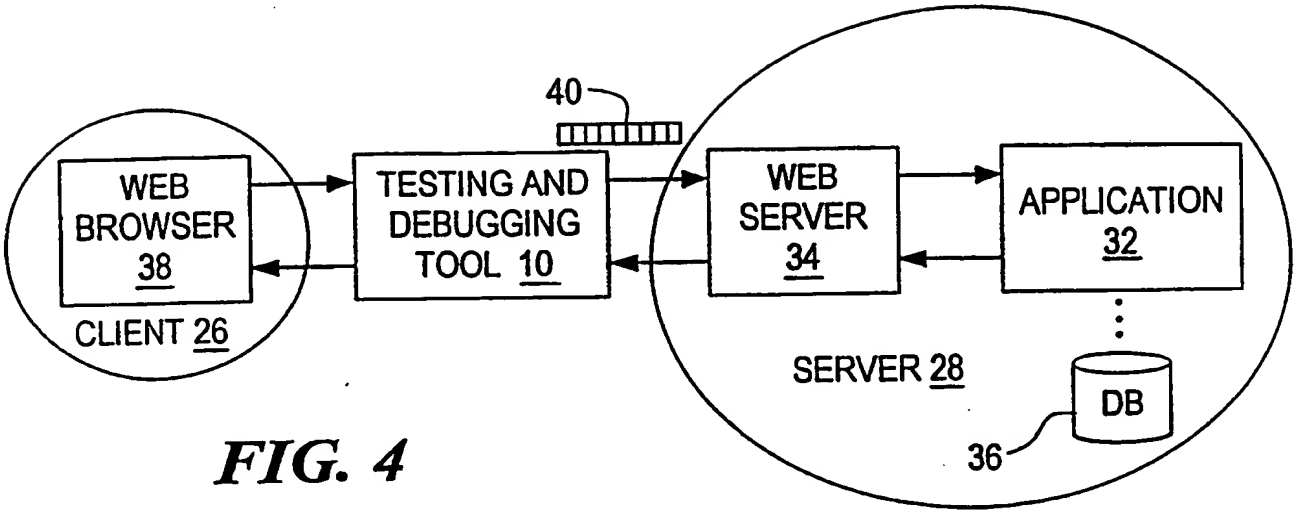


FIG. 4

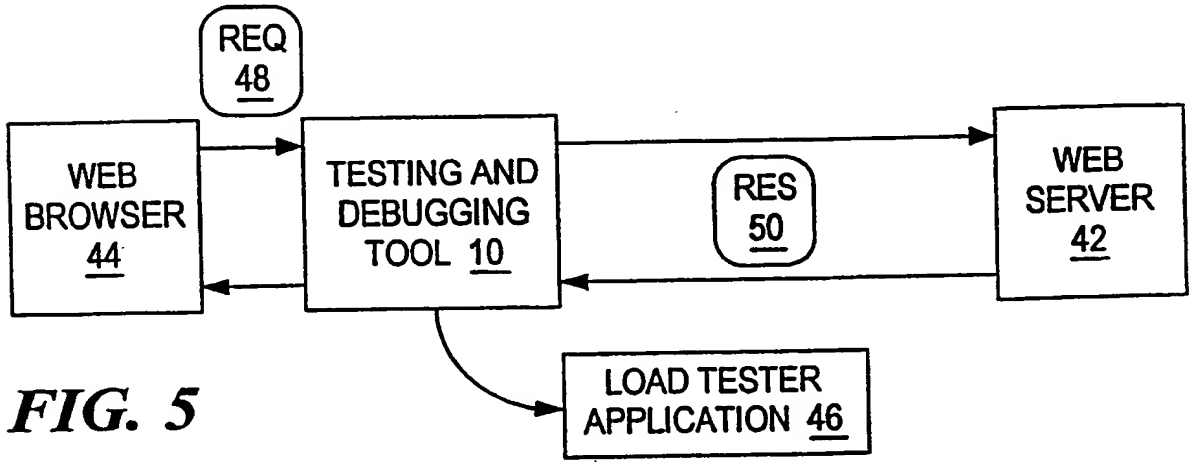


FIG. 5

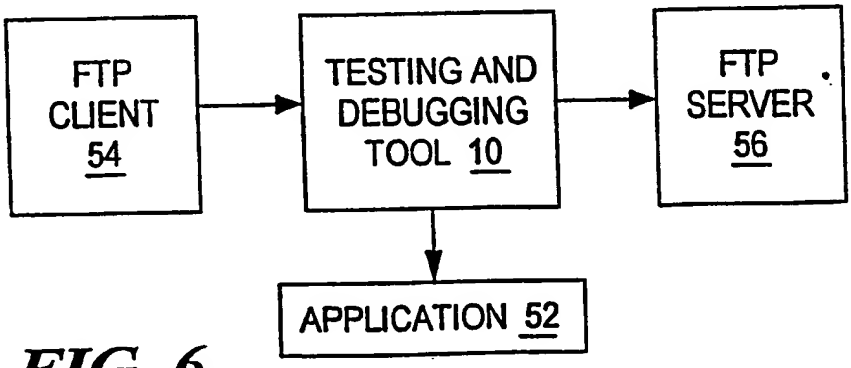


FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/13761

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06F 13/00
US CL :395/200.76,200.33,200.47,200.49
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 395/200.76,200.33,200.47,200.49

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,331,642 A (VALLEY et al) 19 July 1994, col.3, lines 12-48, col. 5-7.	1,6-7,13-14
X,P	US 5,673,322 A (PEPE et al) 30 September 1997, col. 5-6, 10-13.	1-18
Y,P	US 5,727,159 A (KIKINIS) 10 March 1998, col.2-3,6-8.	1-18
A	US 4,720,850 A (OBERLANDER et al) 19 January 1988, see the whole reference.	1-18
A	US 5,706,507 A (SCHLOSS) 06 January 1998, see the whole reference.	1-18
A	US 5,708,654 A (ARNDT et al) 13 January 1998, see the whole reference.	1-18

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 28 AUGUST 1998	Date of mailing of the international search report 16 OCT 1998
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer ZARNI MAUNG Telephone No. (703) 305-3900

Form PCT/ISA/210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/13761

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,774,670 A (MONTULLI) 30 June 1998, see the whole reference.	1-18